



FPT POLYTECHNIC



www.poly.edu.vn

LẬP TRÌNH ANDROID 1

CÁC WIDGET NÂNG CAO TRONG ANDROID

- ❑ Cách sử dụng Spinner, Listview
- ❑ Cách custom Spinner, Listview



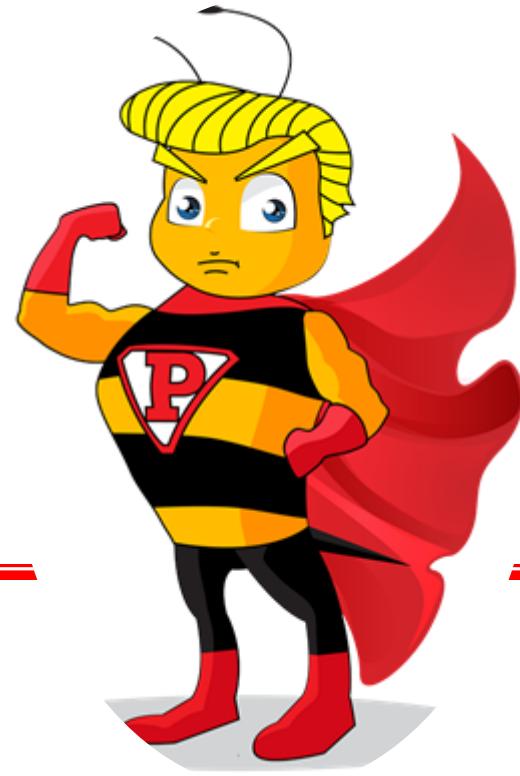
LẬP TRÌNH ANDROID 1

BÀI 5.1: SPINNER VÀ CUSTOM SPINNER

MỤC TIÊU

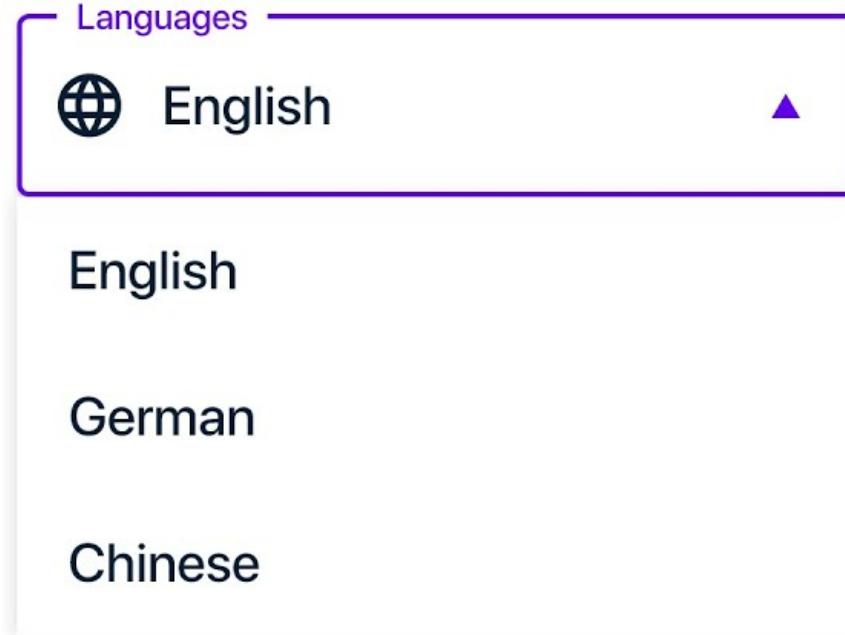
- CÁCH SỬ DỤNG SPINNER, LISTVIEW
- CÁCH CUSTOM SPINNER, LISTVIEW



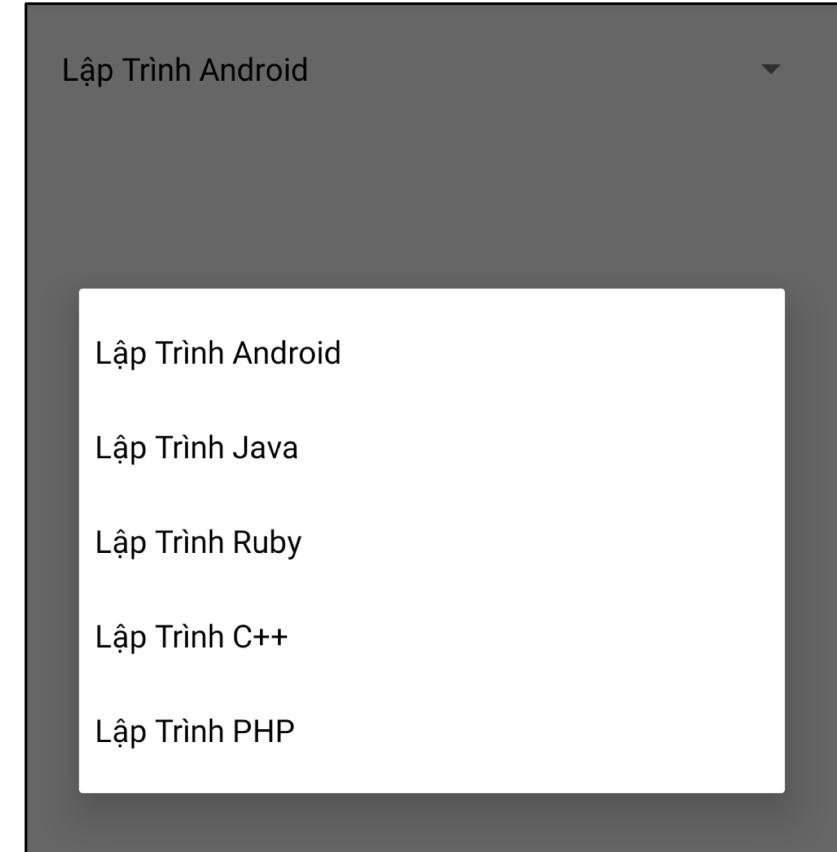
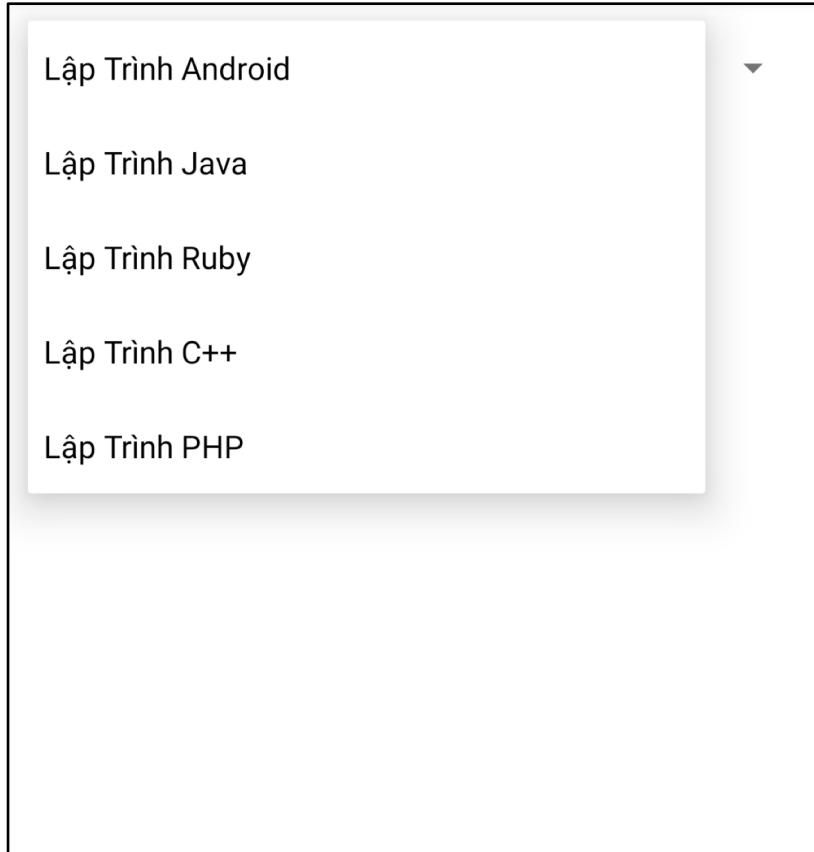


SPINNER VÀ CUSTOM SPINNER

- ❑ **Spinner** là widget trong Android phép bạn chọn một item từ danh sách thả xuống



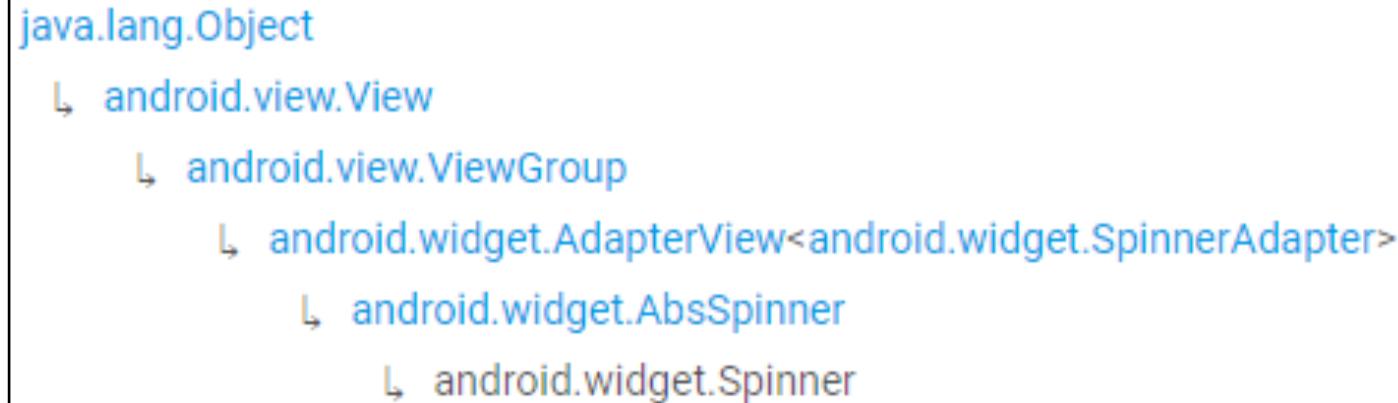
- ❑ Spinner có 2 chế độ (mode) với giao diện khác nhau hoàn toàn:



❖ **android:spinnerMode="dropdown"**

❖ **android:spinnerMode="dialog"**

- ❑ **Spinner** được liên kết với **AdapterView**. Vì vậy cần phải sử dụng một trong các lớp Adapter để *kết nối dữ liệu* cho **Spinner**.
- ❑ Lớp **Spinner** là lớp con của lớp **AbsSpinner**



- ❑ Thông thường một **Spinner** chỉ là chuỗi dữ liệu nên chúng ta chỉ cần sử dụng **ArrayAdapter**. Để hiển thị một Spinner phức tạp hơn gồm hình và chuỗi chúng ta phải tùy biến lại Adapter.

- ❑ Sử dụng **ArrayAdapter** để đổ dữ liệu lên **Spinner**

- ❖ Bước 1: Tạo danh sách dữ liệu cần đổ lên Spinner

```
String[] list = new String[]{  
    "Lập Trình Android",  
    "Lập Trình Java",  
    "Lập Trình Ruby",  
    "Lập Trình C++",  
    "Lập Trình PHP"  
};
```

❑ Sử dụng **ArrayAdapter** để đổ dữ liệu lên **Spinner**

❖ Bước 2: Tạo **ArrayAdapter** và set adapter vừa tạo vào **Spinner**

```
ArrayAdapter<String> spinnerAdapter = new ArrayAdapter<>(  
    this,  
    android.R.layout.simple_spinner_dropdown_item,  
    list  
);  
spinner.setAdapter(spinnerAdapter);
```

❑ Trong ArrayAdapter truyền vào 3 đối số:

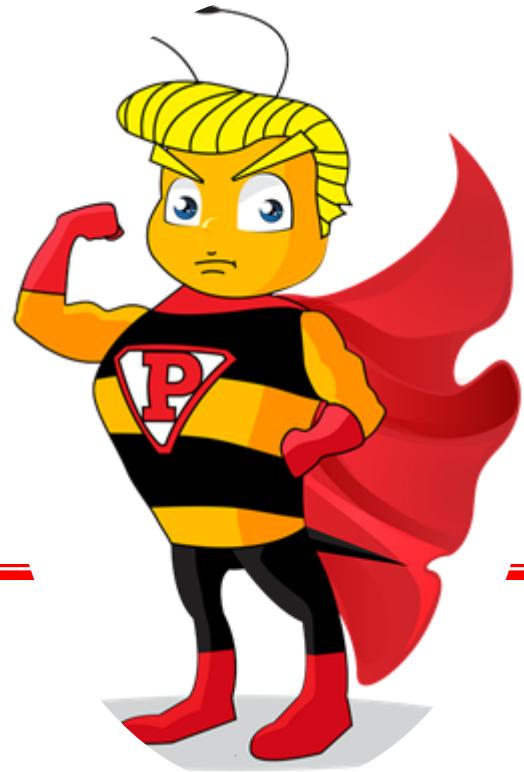
- ❖ Context
- ❖ Layout hiển thị của một item trong spinner, ở đây đang sử dụng **android.R.layout.simple_spinner_dropdown_item** là nguồn tài nguyên có sẵn trong Android
- ❖ Dữ liệu (ở đây là danh sách đã tạo ở bước 1)

Sử dụng **ArrayAdapter** để đổ dữ liệu lên **Spinner**

- ❖ Bước 3: Bắt sự kiện cho Spinner, khi chọn phần tử nào sẽ lấy giá trị của phần tử đó

```
spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
        //phương thức này được gọi khi có một sự kiện chọn item nào đó
        Toast.makeText(MainActivity.this, list[i], Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onNothingSelected(AdapterView<?> adapterView) {
        //phương thức này được gọi khi click vào Spinner mà không chọn item nào cả
    }
});
```



CUSTOM SPINNER

...

❑ Bước 1: Tạo layout cho item hiển thị lên Spinner

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <ImageView
        android:id="@+id/ivAvatar"
        android:layout_width="50dp"
        android:layout_height="50dp" />

    <TextView
        android:id="@+id/txtName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textSize="37sp" />
</LinearLayout>
```

❑ Bước 2: Tạo danh sách dữ liệu cần đổ lên Spinner

```
ArrayList<Employee> list = new ArrayList<>();  
  
list.add(new Employee(1, "Zoe Saldana", R.mipmap.zoe_saldana));  
  
list.add(new Employee(2, "Sam Worthington", R.mipmap.sam_worthington));  
  
list.add(new Employee(3, "Sigourney Weaver", R.mipmap.sigourney_weaver));
```

Ở đây chúng ta đang tạo một danh sách nhân viên, mỗi nhân viên gồm có 3 thông tin ***id, name, avatar***.

- ❑ Bước 3: Tạo một class mới đặt tên là **EmployeeAdapter**, trong class này ta kế thừa **BaseAdapter** và *implement* 4 phương thức: **getCount**, **getItem**, **getItemId**, **getView**

```
public class EmployeeAdapter extends BaseAdapter {  
  
    @Override  
    public int getCount() {  
        return 0;  
    }  
  
    @Override  
    public Object getItem(int i) {  
        return null;  
    }  
  
    @Override  
    public long getItemId(int i) {  
        return 0;  
    }  
  
    @Override  
    public View getView(int i, View view, ViewGroup viewGroup) {  
        return null;  
    }  
}
```

❑ Trong đó:

- ❖ **getCount()** - Phương thức này trả về số dòng của List
- ❖ **getItem(int i)** - Phương thức này trả về Object dựa vào vị trí của đối tượng đó trong danh sách
- ❖ **getItemId(int i)** - Phương thức này trả về id trên View của từng item mà adapter tạo ra.
- ❖ **getView(int i, View view, ViewGroup viewGroup)** - Phương thức này được gọi tự động các mục danh sách để sẵn sàng hiển thị. Trong phương thức chúng ta thiết lập layout cho danh sách các mục bằng cách sử dụng lớp **LayoutInflater** và sau đó thêm dữ liệu cho các view: ImageView, TextView,...

- ❑ Bước 4: Tạo constructor cho EmployeeAdapter với 2 thuộc tính: context và list (danh sách hiển thị lên spinner)

```
private Context context;  
private ArrayList<Employee> list;  
  
public EmployeeAdapter(Context context, ArrayList<Employee> list) {  
    this.context = context;  
    this.list = list;  
}
```

❑ Bước 5: Xử lý các phương thức đã implements trước đó

```
@Override
public int getCount() {
    return list.size();
}

@Override
public Object getItem(int i) {
    return list.get(i);
}

@Override
public long getItemId(int i) {
    return 0;
}

@Override
public View getView(int i, View view, ViewGroup viewGroup) {
    LayoutInflator inflater = ((Activity)context).getLayoutInflator();
    view = inflater.inflate(R.layout.item_spinner, viewGroup, attachToRoot: false);

    ImageView ivAvatar = view.findViewById(R.id.ivAvatar);
    TextView txtName = view.findViewById(R.id.txtName);

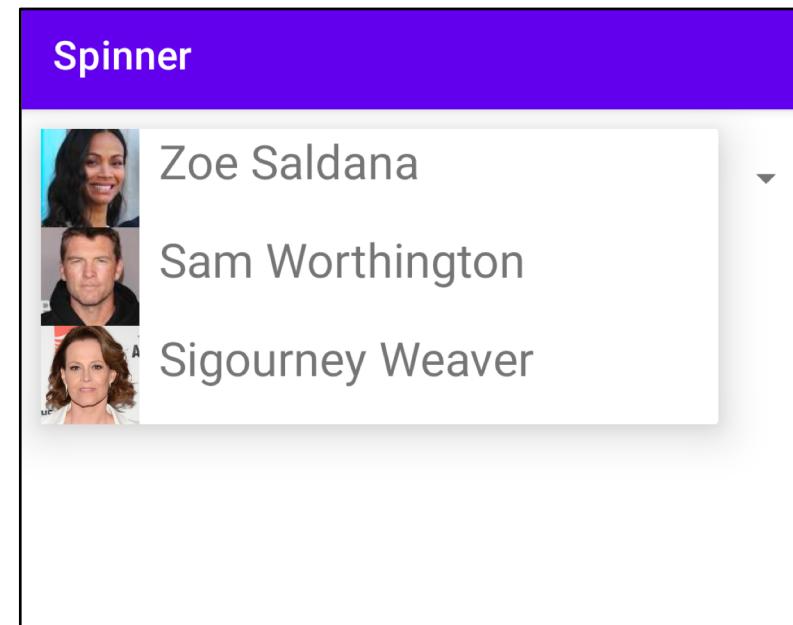
    ivAvatar.setImageResource(list.get(i).getAvatar());
    txtName.setText(list.get(i).getName());

    return view;
}
```

❑ Bước 6: Gán EmployeeAdapter lên Spinner

```
EmployeeAdapter adapter = new EmployeeAdapter(this, list);  
spinner.setAdapter(adapter);
```

❑ Kết quả ta được



- ❑ Bắt sự kiện cho Spinner, khi chọn phần tử nào sẽ lấy giá trị của phần tử đó

```
spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
    @Override
    public void onItemSelected(AdapterView<?> adapterView, View view, int i, long l) {
        //phương thức này được gọi khi có một sự kiện chọn item nào đó
        Toast.makeText(MainActivity.this, list.get(i).getName(), Toast.LENGTH_SHORT).show();
    }

    @Override
    public void onNothingSelected(AdapterView<?> adapterView) {
        //phương thức này được gọi khi click vào Spinner mà không chọn item nào cả
    }
});
```



FPT POLYTECHNIC



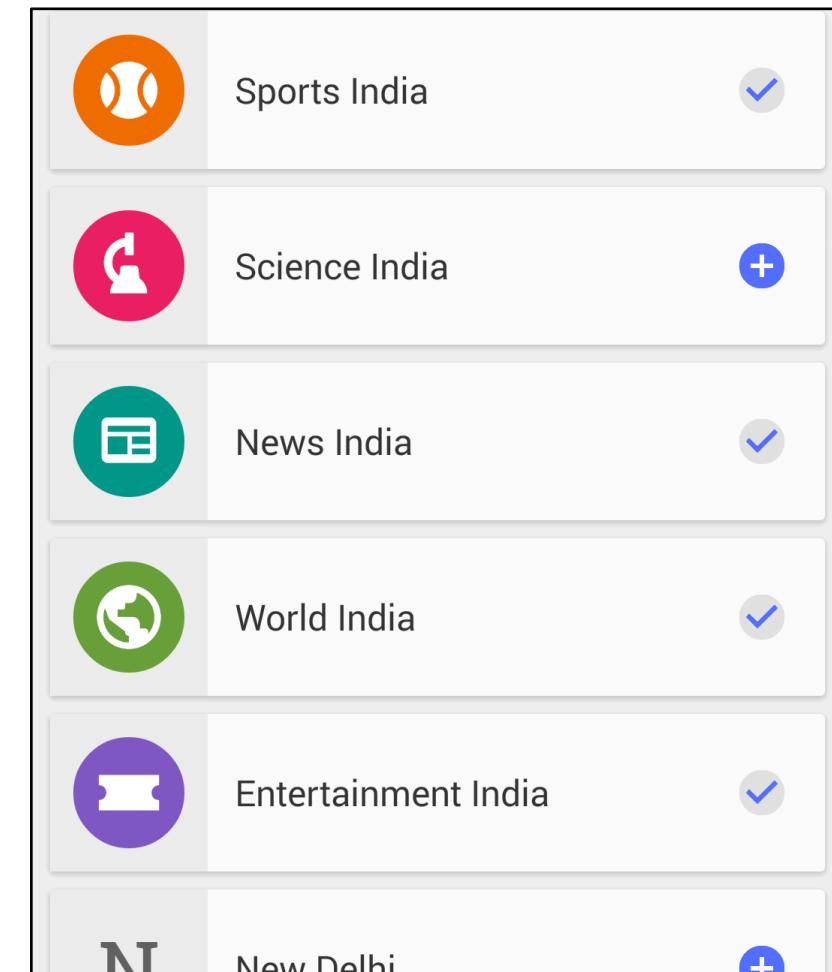
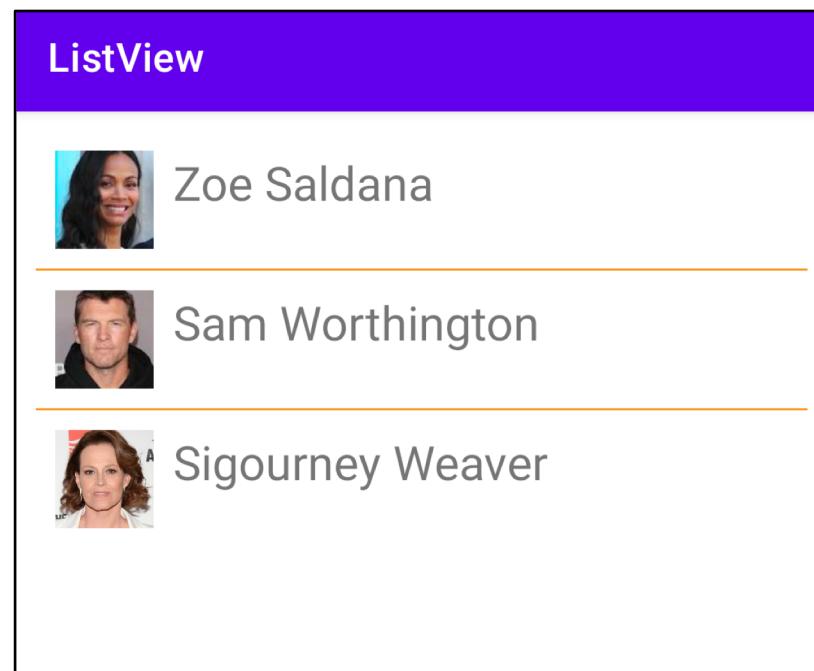
www.poly.edu.vn

BÀI 5.2: LISTVIEW & CUSTOM LISTVIEW



LISTVIEW & CUSTOM LISTVIEW

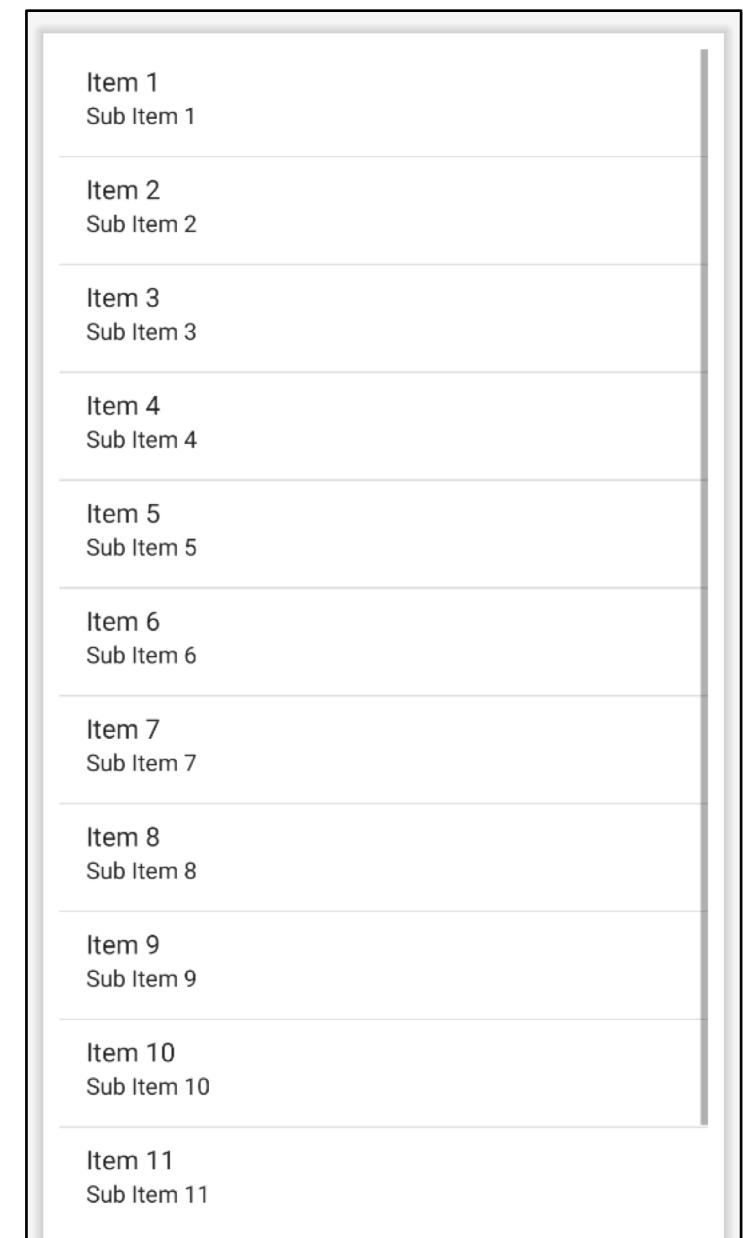
- ❑ **ListView** là một view chứa nhóm các mục và hiển thị trong một danh sách có thể cuộn được. Người sử dụng có thể chọn item trong danh sách bằng cách Click vào item đó



☐ ListView code trong XML

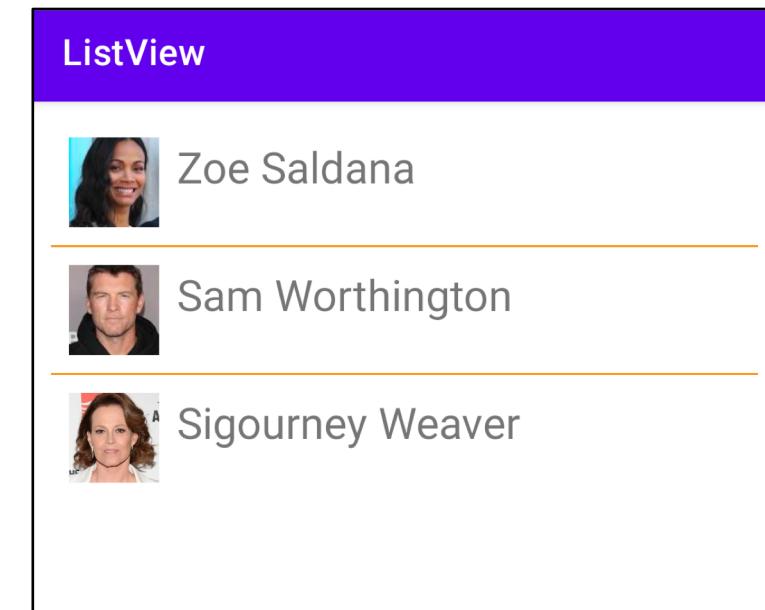
<ListView

```
    android:id="@+id/lvEmployee"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```



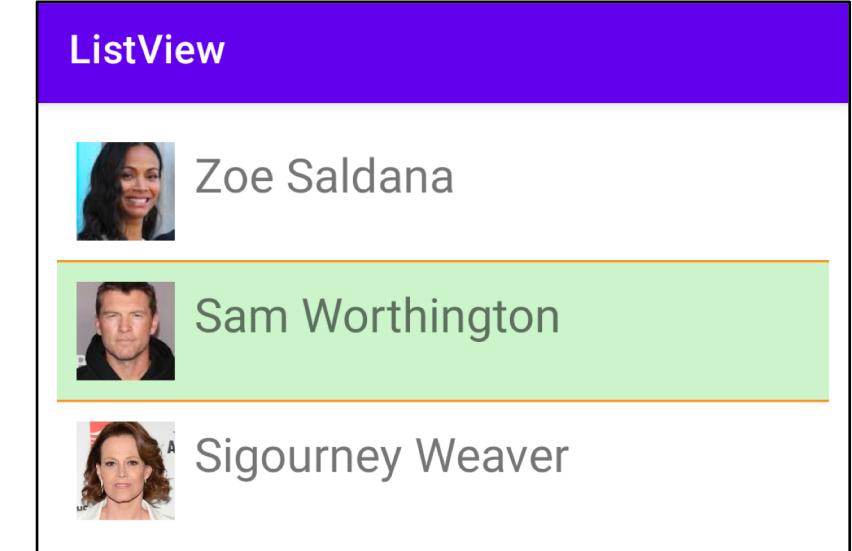
- ❑ **android:divider** - Thuộc tính này có thể là một image hay màu dùng để phân chia giữa các dòng trong ListView
- ❑ **android:dividerHeight** - Thuộc tính này xác định chiều cao của thuộc tính *android:divider*

```
<ListView  
    android:id="@+id/lvEmployee"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:divider="#FA961E"  
    android:dividerHeight="1dp"/>
```



- ❑ **android:listSelector** - Thuộc tính này thường được sử dụng để thiết lập màu hoặc hình dòng được chọn trong listView

```
<ListView  
    android:id="@+id/lvEmployee"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:divider="#FA961E"  
    android:dividerHeight="1dp"  
    android:listSelector="#CBF4CB"/>
```



❑ Bước 1: Tạo danh sách dữ liệu cần đổ lên **Listview**

```
ArrayList<String> list = new ArrayList<>();  
  
list.add("Pizza");  
  
list.add("Hot dog");  
  
list.add("Sandwich");  
  
list.add("Phở");  
  
list.add("Bánh mì");
```

❑ Bước 2: Tạo ArrayAdapter và set adapter vừa tạo lên Listview

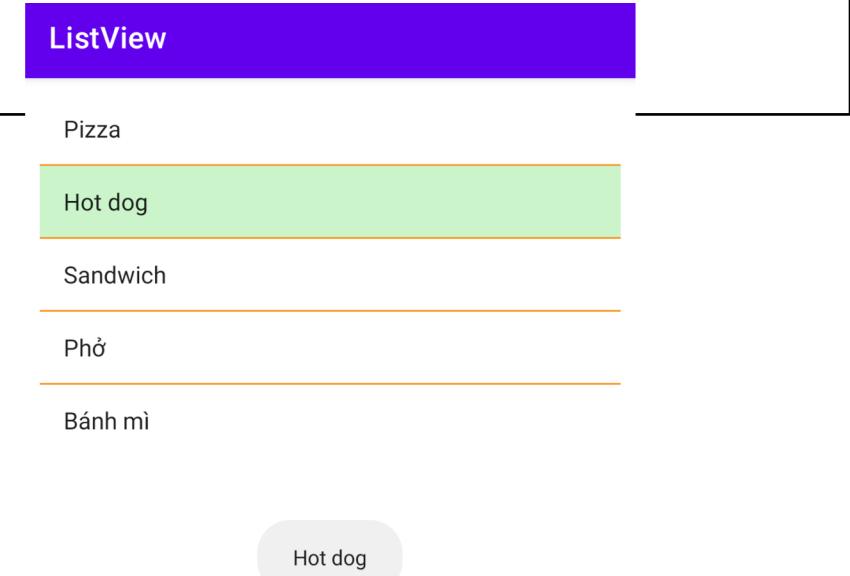
```
ArrayAdapter<String> adapter = new ArrayAdapter<>(  
    this,  
    android.R.layout.simple_list_item_1,  
    list  
);  
lvFood.setAdapter(adapter);
```

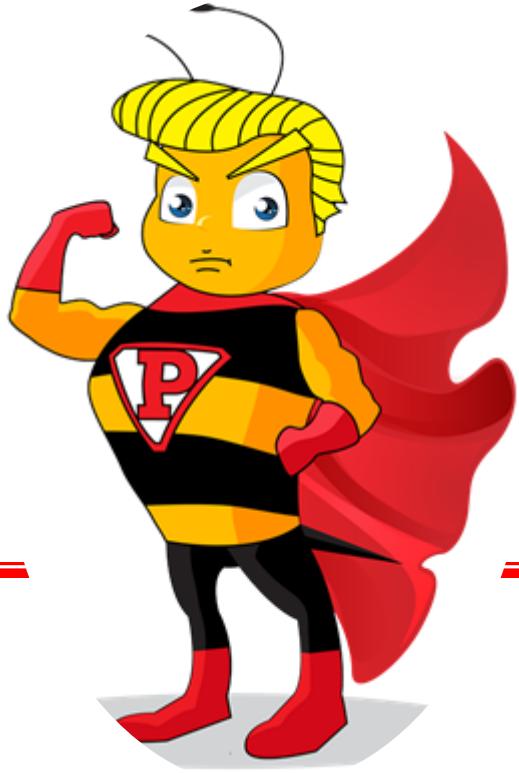
❑ Trong ArrayAdapter truyền vào 3 đối số:

- ❖ Context
- ❖ Layout hiển thị của một item trong spinner, ở đây đang sử dụng **android.R.layout.simple_list_item_1** là nguồn tài nguyên có sẵn trong Android
- ❖ Dữ liệu (ở đây là danh sách đã tạo ở bước 1)

- ❑ Bước 3: Bắt sự kiện cho **Listview**, khi chọn phần tử nào sẽ lấy giá trị của phần tử tương ứng

```
lvFood.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
        Toast.makeText(MainActivity.this, list.get(i), Toast.LENGTH_SHORT).show();  
    }  
});
```





CUSTOM LISTVIEW

❑ Bước 1: Tạo layout cho item hiển thị lên ListView

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="10dp">

    <ImageView
        android:id="@+id/ivFood"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:scaleType="fitXY"/>

    <TextView
        android:id="@+id/txtName"
        android:layout_width="match_parent"
        android:layout_height="50dp"
        android:layout_marginStart="10dp"
        android:textSize="24sp" />
</LinearLayout>
```

❑ Bước 2: Tạo danh sách dữ liệu cần đổ lên ListView

```
ArrayList<Food> list = new ArrayList<>();  
  
list.add(new Food(1, "Pizza", R.mipmap.pizza));  
  
list.add(new Food(2, "Phở", R.mipmap.pho));  
  
list.add(new Food(3, "Bánh mì", R.mipmap.banh_mi));
```

Ở đây chúng ta đang tạo một danh sách món ăn, mỗi món ăn gồm có 3 thông tin ***id, name, image***.

- ❑ Bước 3: Tạo một class mới đặt tên là **FoodAdapter**, trong class này ta kế thừa **BaseAdapter** và *implement* 4 phương thức: **getCount**, **getItem**, **getItemId**, **getView**

```
public class FoodAdapter extends BaseAdapter {

    @Override
    public int getCount() {
        return 0;
    }

    @Override
    public Object getItem(int i) {
        return null;
    }

    @Override
    public long getItemId(int i) { return 0; }

    @Override
    public View getView(int i, View view, ViewGroup viewGroup) {
        return null;
    }
}
```

❑ Trong đó:

- ❖ **getCount()** - Phương thức này trả về số dòng của List
- ❖ **getItem(int i)** - Phương thức này trả về Object dựa vào vị trí của đối tượng đó trong danh sách
- ❖ **getItemId(int i)** - Phương thức này trả về id trên View của từng item mà adapter tạo ra.
- ❖ **getView(int i, View view, ViewGroup viewGroup)** - Phương thức này được gọi tự động các mục danh sách để sẵn sàng hiển thị. Trong phương thức chúng ta thiết lập layout cho danh sách các mục bằng cách sử dụng lớp **LayoutInflater** và sau đó thêm dữ liệu cho các view: ImageView, TextView,...

- ❑ Bước 4: Tạo constructor cho FoodAdapter với 2 thuộc tính: context và list (danh sách hiển thị lên Listview)

```
private Context context;  
private ArrayList<Food> list;  
  
public FoodAdapter(Context context, ArrayList<Food> list) {  
    this.context = context;  
    this.list = list;  
}
```

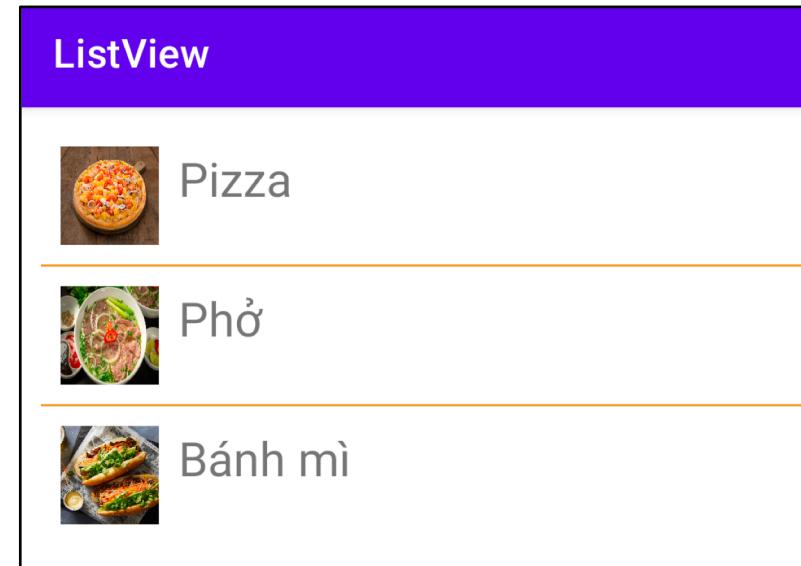
❑ Bước 5: Xử lý các phương thức đã implements trước đó

```
@Override  
public int getCount() { return list.size(); }  
  
└ Dinh Nguyen  
@Override  
public Object getItem(int i) { return list.get(i); }  
  
└ Dinh Nguyen  
@Override  
public long getItemId(int i) { return 0; }  
  
└ Dinh Nguyen *  
@Override  
public View getView(int i, View view, ViewGroup viewGroup) {  
    LayoutInflator inflater = ((Activity) context).getLayoutInflater();  
    view = inflater.inflate(R.layout.item_listview, viewGroup, attachToRoot: false);  
  
    ImageView ivAvatar = view.findViewById(R.id.ivFood);  
    TextView txtName = view.findViewById(R.id.txtName);  
  
    ivAvatar.setImageResource(list.get(i).getFood());  
    txtName.setText(list.get(i).getName());  
  
    return view;  
}
```

❑ Bước 6: Gán FoodAdapter lên ListView

```
FoodAdapter adapter = new FoodAdapter(this, list);  
lvFood.setAdapter(adapter);
```

❑ Kết quả ta được



CUSTOM LISTVIEW (BASEADAPTER)

- ☐ Bắt sự kiện cho ListView, khi chọn phần tử nào sẽ lấy giá trị của phần tử đó

```
lvFood.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {  
        Toast.makeText(MainActivity.this, list.get(i).getName(), Toast.LENGTH_SHORT).show();  
    }  
});
```

ListView



Pizza



Phở



Bánh mì

Phở



FPT Education

FPT POLYTECHNIC

Thank you